applied to candidate word scores. That is, the score of a candidate word is multiplied by the usage frequency ranking of the candidate word, and candidate words for presentation are selected based on their weighted scores.

[0075] As another example, another scheme replaces candidate word scoring based on character-by-character comparisons, as described above, with scoring based on the edit distance (also known as the Levenshtein distance) between the input sequence and the candidate word. That is, the score of a candidate word is the edit distance between the candidate word and the input sequence, or a function thereof, and candidate words are selected for presentation based on the edit distance scores. Alternately, the score for each candidate is based on the edit distance multiplied by (or otherwise combined with) the usage frequency ranking of the candidate, and candidate words are selected for presentation based on these scores.

[0076] As another example, another scheme uses a graph-matching technique. In this technique, the sequence of individual touch points that a user inputs into the device for a word (e.g., by contacts with a virtual keyboard on the touch screen) form a directed graph. This user-input directed graph is compared against a collection of directed graphs for respective words in a dictionary to generate a list of dictionary words that most closely match the user typing. In some embodiments, the probability that a user-input directed graph matches the directed graph for a dictionary word is calculated as follows:

[0077] Let $U_{1 \ldots n}$ be each point in the user-input directed graph.

[0078] Let $D_{1 \ldots n}$ be each point in the directed graph of a dictionary word. Points in this directed graph are assigned based on the centroid of the key that inputs the corresponding letter, as represented in the keyboard user interface.

[0079] Let $P_{1 \ldots n}$ be, for each point in the user-input directed graph, the probability that the letter corresponding to U_x equals the letter corresponding to D_x. In some embodiments, a respective P_x is computed by calculating the Euclidean distance between the points U_x and D_x, and applying a factor based on the size of the user interface elements that indicate the keys on the keyboard. A minimum probability may be entered for P_x if the graphs for the user word and the dictionary word are different lengths. In one embodiment, the factor (based on the size of the user interface elements that indicate the keys on the keyboard) is a divisor that is equal to, or proportional to, the distance between center points of two horizontally adjacent keys on the keyboard.

[0080] Multiplying the probabilities in $P_{1 \ldots n}$ together yields G, the probability that a graph for a dictionary word matches the user-input graph. In some embodiments, G is multiplied by F, the frequency that the word occurs in the source language/domain. Furthermore, in some embodiments G is also multiplied by N, a factor calculated by considering one or more words previously typed by the user. For example, in a sentence/passage being typed by a user, "to" is more likely to follow "going," but "ti" is more likely to follow "do re mi fa so la." In some embodiments, G is multiplied by both F and N to yield Ω, the probability that a user-input directed graph matches a dictionary word.

[0081] The collection of dictionary words with the highest probabilities may be presented in a display for user consideration, for example as described in "Method, System, and Graphical User Interface for Providing Word Recommendations" (U.S. Patent Application number to be determined, filed Jan. 5, 2007, attorney docket number 063266-5041), the content of which is hereby incorporated by reference in its entirety. In other cases, the top-ranked word is selected for the user by the device without user intervention.

[0082] In some embodiments, as word recommendations are offered by the portable device and selected by the user, statistics regarding the corrections made are collected. For example, the characters in an input sequence that was replaced by a candidate word selected by the user and the corresponding characters may be logged. Over time, the corrections log may be analyzed for patterns that may indicate a pattern of repeated typing errors by the user. If the keyboard is a virtual keyboard on a touch screen of the portable device, the portable device may automatically adjust or recalibrate the contact regions of the keys of the virtual keyboard to compensate for the user pattern of typing errors. As another example, for a given input sequence, the word selected by the user may be recommended first or given a higher score when the same input sequence is subsequently entered by the user.

[0083] The foregoing description, for purpose of explanation, has been described with reference to specific embodiments. However, the illustrative discussions above are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated.

What is claimed is:

1. A computer-implemented method, comprising:

receiving a sequence of input characters from a keyboard, wherein the keyboard has a predefined layout of characters with each character in the layout having one or more neighbor characters;

generating a set of strings from at least a subset of the sequence of input characters, the set of strings comprising permutations of respective input characters in the subset of the sequence and neighbor characters of the respective input characters on the layout of the keyboard;

identifying in a dictionary one or more candidate words, each candidate word having a string in the set of strings as a prefix;

scoring the candidate words;

selecting a subset of the candidate words based on predefined criteria; and

presenting the subset of the candidate words.

2. The method of claim **1**, wherein scoring a respective candidate word comprises:

comparing a respective character in each character position of the candidate word with a respective character in a corresponding position in the sequence of input characters; and

determining a score for the respective candidate word based on the comparing.

3. The method of claim **2**, wherein scoring the respective candidate word further comprises increasing the score of the respective candidate word if the respective candidate word, compared to the sequence of input characters, has only one character that is different.

4. The method of claim **2**, wherein scoring the respective candidate word further comprises increasing the score of the